Snapshot Stats

Kate Lance
clance@connect.com.au

The amount of information that a sysadmin must personally filter is immense, and falls into many different system areas such as user activity, security, emergencies, resources, and performance. There are many fine utilities available to sound the alarm when resources collapse, emergencies errupt, or security is attacked.

But how do you get a feel for the everyday performance of your machines, and how do you find out when their resources are gradually drifting away from optimal levels to the point at which you need to re-provision them? Performance indicators such as free memory, available CPU, process load, or disk activity may take months before functional levels degrade to potential bottlenecks, and the baseline changes may be so gradual that they're lost in the noise of often enormous daily fluctuations.

The everyday performance tools like ***vmstat*** or ***ping*** give instantaneous pictures of the state of a machine or network, but they demand constant monitoring to perceive changes -- and how do you tell if they're significant changes or just acceptable normal fluctuations?

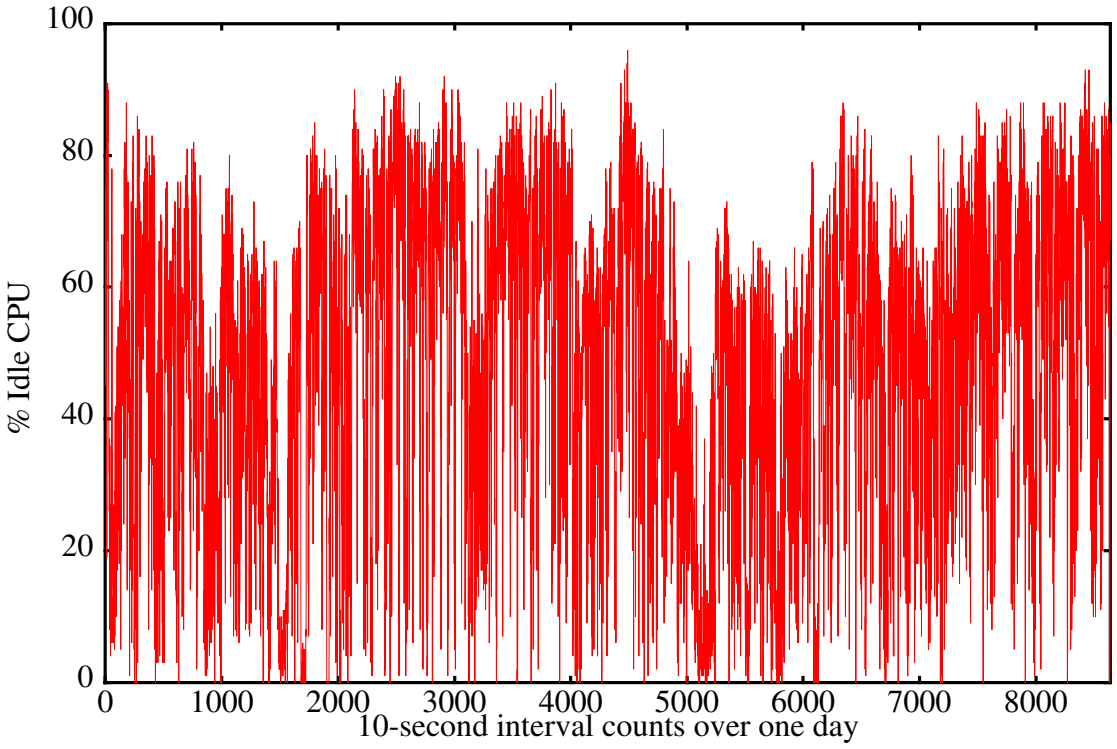

Fig. 1: Percentage of idle CPU measured at 10-second intervals over one day on a busy News server.

You could collect the data continuously for later inspection, but this would not only impact on performance, but such a vast stream of data would be very difficult to summarise, understand, compare, or preserve for future investigation.

The fairly obvious answer is to take samples of the data, not to try to collect it continually. If samples are appropriately spaced then they provide a snapshot of the conditions of machines or networks. But this raises two questions:

- what are the *appropriate* sample intervals for targets with such enormously variability?
- and once you have identified a useful snapshot interval how do you comprehend the data on a regular basis to know when a *significant* change has occurred?

Appropriate Sample Intervals

To define appropriate sample intervals, the utility ***vmstat*** was run every 10 seconds for one full day on a very active News server, with great variability in short and long time-scales throughout the day:

procs			memory		page						disk				faults			cpu		
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	s0	s1	s2	s3	in	sy	cs	us	sy	id
0	0	0	430784	4216	0	90	14	0	0	0	0	1	2	1	8	314	1662	292	9	16	75
0	0	0	431212	3832	2	202	17	12	56	360	20	0	2	0	21	326	1645	299	11	23	66
1	0	0	428580	3188	14	326	28	91	434	732	160	1	7	3	39	438	1852	338	22	33	45
0	0	0	430180	4816	1	230	59	8	8	0	0	2	4	2	24	403	2207	342	16	25	59
0	0	0	429540	3400	3	135	22	13	13	0	0	1	3	3	14	318	1732	279	9	20	71
1	1	0	429716	3544	21	260	14	88	88	0	0	1	6	6	61	498	1968	348	17	32	52
1	1	0	429408	3676	24	263	28	121	121	0	0	0	5	2	69	511	2389	398	29	35	36
0	0	0	430332	4088	1	77	18	0	0	0	0	0	3	1	45	392	27871	286	21	60	19
0	0	0	430884	5424	0	64	17	0	0	0	0	0	1	1	5	291	1697	284	5	14	81

The mean of each of the measurements from a full day's data was calculated and compared to the mean derived from samples taken at one-minute, 5-minute, 10, 15, 20, 30 60 and 120-minute intervals:

Number of samples over one day	Sample interval (minutes)
8640	1/6
1440	1
288	5
144	10
96	15
72	20
48	30
24	60
12	120

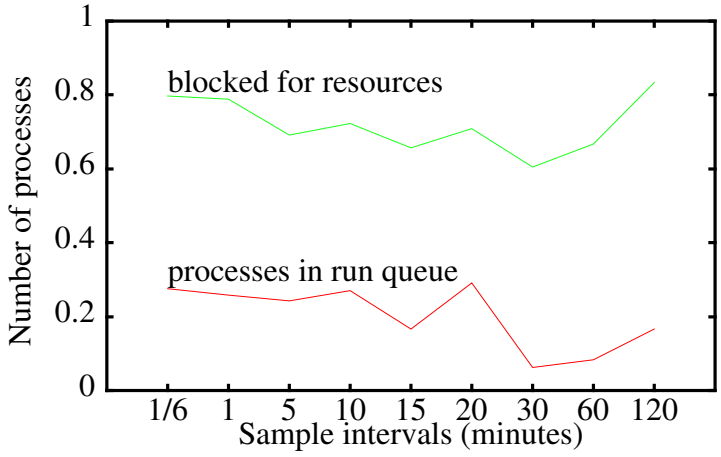

Fig. 2: Mean number of processes in run queue and blocked for resources for each of the sample intervals (procs, r b).

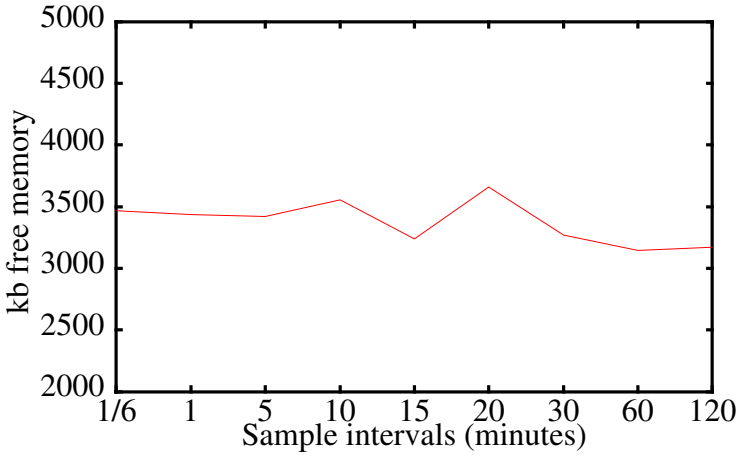

Fig. 3: Mean kbytes free memory for each of the sample intervals (memory, free).

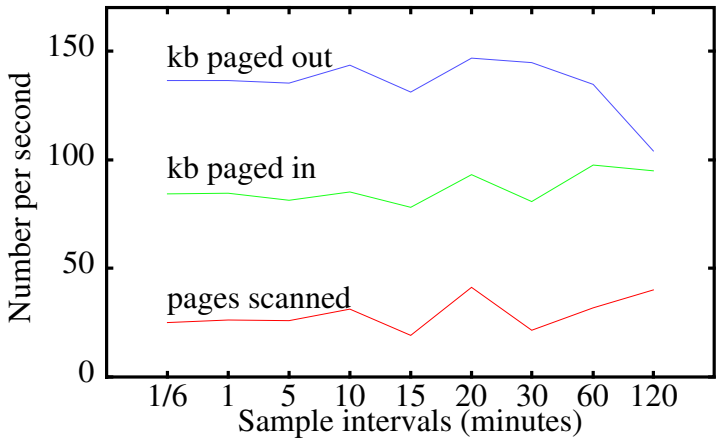

Fig. 4: Mean kb paged in, out and pages scanned for each of the sample intervals (pi, po, sr).

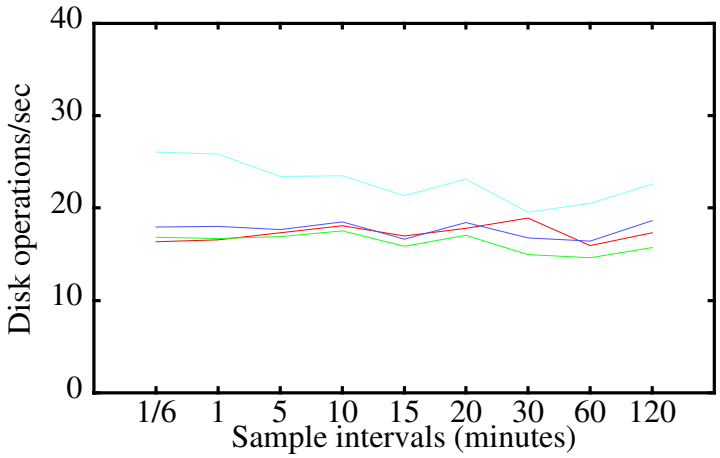

Fig. 5: Mean number of disk operations per second for 4 disks for each of the sample intervals (disk, s0 s1 s2 s3).

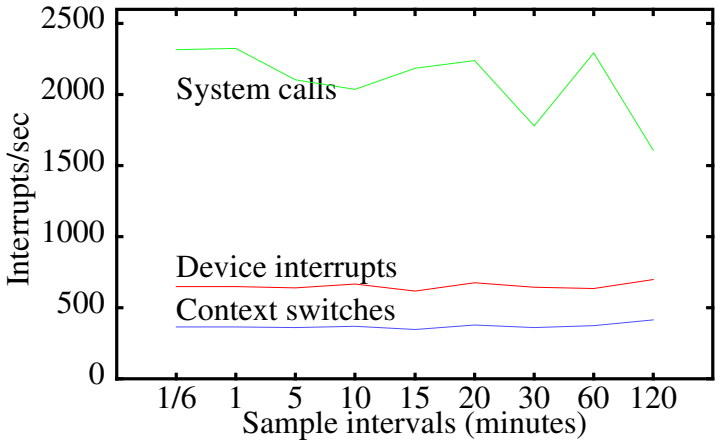

Fig. 6: Mean interrupts per second for each of the sample intervals (faults, in sy cs).

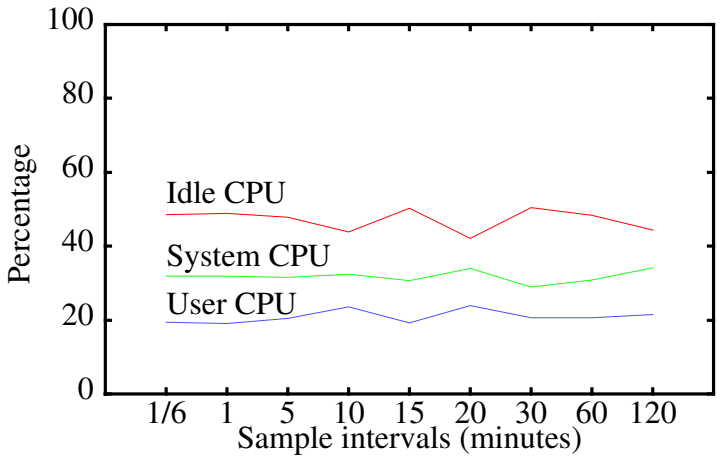

Fig. 7: Mean percentage of Idle, System and User CPU for each of the sample intervals (cpu, us sy id).

- The surprising result is that samples taken as infrequently as once per hour or once every two hours give mean results that are almost as accurate as those taken once every 10 seconds.

- 10, 15, 20, and 30 minute intervals show large differences from 10-second interval data -- perhaps cron activity at those intervals is affecting the means.

Recognising Significant Changes

Our procedure:

- Collect data (hourly at our site), save to files with a datestamp, one flat file per machine -- can plot daily fluctuations easily (gnuplot). Files rolled over monthly.

- At midnight a simple shell script takes means of the previous 24 hours of data, emails system staff a list of machines and their mean results for the most informative of the above measures:

Processes in run queue
Blocked processes
Page scan rate
System call interrupts
Idle CPU

Comparison *between* machines:

```
Machine           Load   Block    Scan   Inter   freeCPU
perki              0.1     0.3      74    1681    71.3     (News servers)
merki              0.4     0.2      50    1297    56.3
gnamma             1.4     1.5      32     278    31.2
yeppa              1.6     1.2      51     382    32.9
auwa               0.4     1.0      75     233    63.0
pinah              0.2     0.3      18     159    74.1

myangup            0.3     0.9      11   10447    40.2     (Web caches)
myeah              0.3     0.7       7   11810    35.8
myponga            0.1     0.6      10   11064    42.7

bunora             0.0     0.1       3    1096    71.4     (VWS servers)
buntor             0.0     0.1       0     477    93.3
bunyip             0.3     0.1       0     587    71.2
```

- News servers: high loads, blocked resources (usually disk I/O), high scan rates (need memory), low interrupt rates, two are low on available CPU.

- Web caches: occasional load, some disk I/O blockage, memory starting to run low, very high interrupt rates (but machines are Ultra 2's, can handle it), but CPU resources low.

- VWS servers: low load, little disk I/O, good memory resources, low interrupt rate, lot of CPU available.

Comparison between daily results for *each* machine:

- We save the daily mean results to a flat file for each machine, for easy scanning by eye or plotting.

Before and after memory, CPU upgrade, disk reorganisation:

```
# UnivTime                Load   Block    Scan   Inter   freeCPU
# Sun Mar 9 00:00:06 EST 1997
857908681                  0.0     1.4      63    5788    71.8
857995081                  0.0     1.5      56    6045    68.5
858081481                  0.1     2.3      77   12620    56.4
858167880                  0.2     2.0      81   26265    41.4
858254281                  0.1     1.7      61   25959    43.0
858340682                  0.1     3.4      99   22532    42.6
858427081                  0.1     3.0      71   26184    42.7
# Sun Mar 16 00:00:06 EST 1997
858513481                  0.0     2.0      76   31239    41.5
858599881                  0.1     3.5      87   10642    60.8
858686280                  0.1     2.2      94    7264    61.4 *****
858772681                  0.1     1.4     103    1728    69.3
858859082                  0.0     0.8      75    1622    72.4
858945480                  0.0     0.3      74    1951    65.7
859031881                  0.0     0.6      53    1124    76.6
```

Runaway process that consumed all CPU:

```
863359081                0.0      0.0        1      292    93.7
863445482                0.0      0.0        0    15090    41.2
863531881                0.0      0.1        0    26897     0.0 *****
863618281                0.0      0.0        3    12522    52.3
863704682                0.0      0.0        1      560    92.9
863791082                0.0      0.0        0      395    93.3
```

- The actual figures are much less important than **changes** in the figures.
- Can't exclude human involvement! You must *know* your machine to start with, know what sort of services it's providing and what are normal, acceptable performance indicators for that machine -- will vary according to CPU type, amount of memory, disk layout, services performed.
- Over time, once a pattern of activity is established, you could set threshold values, get email when they are exceeded, etc. -- but I prefer, just once a day, to scan a familiar list and see for myself any differences.
- When funds becomes available for upgrades or extra memory, it is obvious from daily patterns like this which boxes really need the resources.
- The tiny flat files are easy to store, easy to scan by eye, easy to plot for long-term trends, easy to refer to in the future if some other aspect of the information becomes interesting.

We use snapshot stats techniques with:
- ***vmstat*** for monitoring machine performance and general health (as above)
- ***ping*** for core network machine response
- ***ping*** for checking congestion and response times on international links to popular sites
- ***ping*** for path-specific checks on our international bandwidth providers
- ***url_get*** for measuring response of overseas and internal web sites from our Sydney and Melbourne hubs